

matlab 实现基于 PCA 的人脸识别算法

实验者: guodw3

一、问题描述

在一个 yale 人脸库中, 有 15 个人, 每人有 11 幅图像。要求选定每一个人的若干幅图像组成样本库, 由样本库得到特征库。再任取 yale 图像库的一张图片, 识别它的身份。

对于一幅图像可以看作一个由像素值组成的矩阵, 也可以扩展开, 看成一个矢量。如一幅 $N \times N$ 像素的图像可以视为长度为 N^2 的矢量, 这样就认为这幅图像是位于 N^2 维空间中的一个点, 这种图像的矢量表示就是原始的图像空间, 但是这个空间仅是可以表示或者检测图像的许多个空间中的一个。不管子空间的具体形式如何, 这种方法用于图像识别的基本思想都是一样的, 首先选择一个合适的子空间, 图像将被投影到这个子空间上, 然后利用对图像的这种投影间的某种度量来确定图像间的相似度, 最常见的就是各种距离度量。因此, 本次采用 PCA 算法确定一个子空间, 最后使用最小距离法进行识别, 并用 matlab 实现。

二、PCA 原理和人脸识别方法

1) K-L 变换

K-L 变换以原始数据的协方差矩阵的归一化正交特征矢量构成的正交矩阵作为变换矩阵, 对原始数据进行正交变换, 在变换域上实现数据压缩。它具有去相关性、能量集中等特性, 属于均方误差测度下, 失真最小的一种变换, 是最能去除原始数据之间相关性的一种变换。

PCA 则是选取协方差矩阵前 k 个最大的特征值的特征向量构成 K-L 变换矩阵。

2) 主成分的数目的选取

保留多少个主成分取决于保留部分的累积方差在方差总和中所占百分比(即累计贡献率), 它标志着前几个主成分概括信息之多寡。实践中, 粗略规定一个百分比便可决定保留几个主成分; 如果多留一个主成分, 累积方差增加无几, 便不再多留。

3) 人脸空间建立

假设一幅人脸图像包含 N 个像素点, 它可以用一个 N 维向量 Γ 表示。这样, 训练样本库就可以用 Γ_i ($i=1, \dots, M$) 表示。协方差矩阵 C 的正交特征向量就是组成人脸空间的基向量, 即特征脸。

将特征值由大到小排列: $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_r$, 其对应的特征向量为 μ_k 。这样每一幅人脸图像都可以投影到由 u_1, u_2, \dots, u_r 张成的子空间中。因此, 每一幅人脸图像对应于子空间中的一点。同样, 子空间的任意一点也对应于一幅图像。

4) 人脸识别

有了这样一个由“特征脸”张成的降维子空间，任何一幅人脸图像都可以向其投影得到一组坐标系数，这组系数表明了该图像在子空间中的位置，从而可以作为人脸识别的依据。计算数据库中每张图片在子空间中的坐标，得到一组坐标，作为下一步识别匹配的搜索空间。

计算新输入图片在子空间中的坐标，采用最小距离法，遍历搜索空间，得到与其距离最小的坐标向量，该向量对应的人脸图像即为识别匹配的结果。

三、实验步骤

- 1) 每人选取 4 幅共 60 幅作为训练样本，将每一幅图像（128*128）写成列向量形式 X_i （16384*1）排列成数据矩阵

$$X = (X_1, X_2, \dots, X_n) \quad (n=60)$$

- 2) 求均值向量

$$\mu = \frac{1}{n} \sum_{i=1}^n X_i$$

中心化后数据矩阵

$$C = (X_1 - \mu, X_2 - \mu, \dots, X_n - \mu)$$

协方差矩阵

$$\Sigma = \frac{1}{n} C C^T$$

- 3) 通过求 $C^T C$ 来求协方差矩阵特征值，选取最大的 k 个，求出特征向量（特征脸） e_i ，将 k 个这样的特征向量按列排成变换矩阵（16384*k）

$$W = (e_1, e_2, \dots, e_k)$$

- 4) 计算每幅图像的投影（ k 维列向量）

$$Y_i = W^T (X_i - \mu)$$

- 5) 计算待识别人脸的投影（ k 维列向量），设待识别人脸为 Z

$$chZ = W^T (Z - \mu)$$

- 6) 遍历搜索进行匹配

$$Y_j = \min ||Y_i - chZ||$$

那么 Z 就是第 j 个人

四、实验结果与分析

matlab 界面效果如下所示

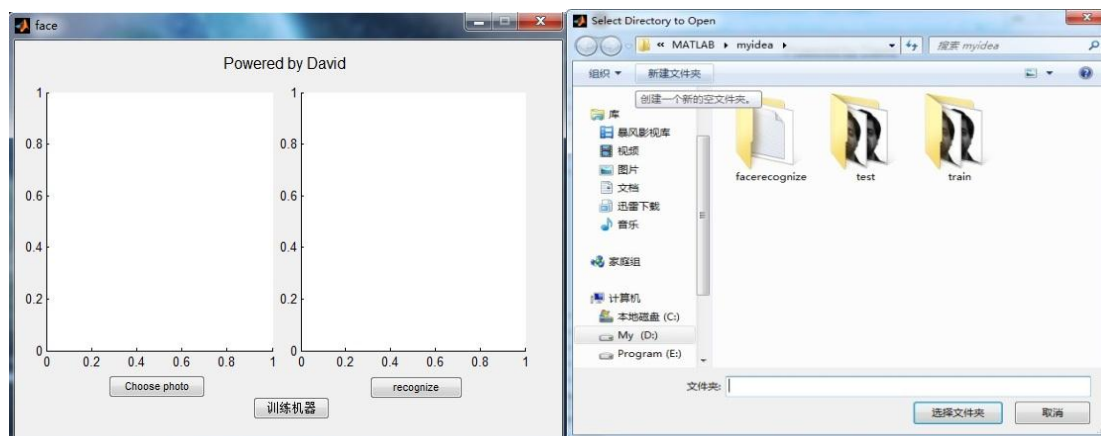


图 1 用户使用界面



图 2 选择图片

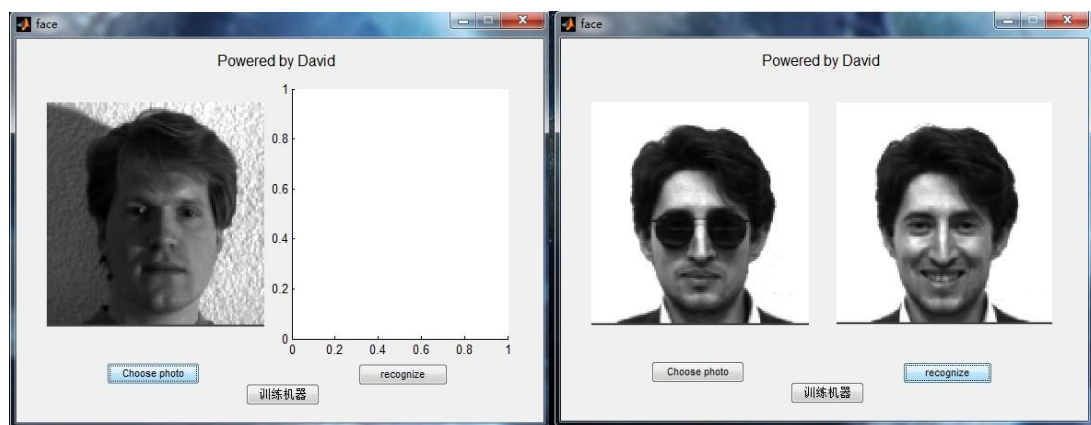


图 3 图片选择后

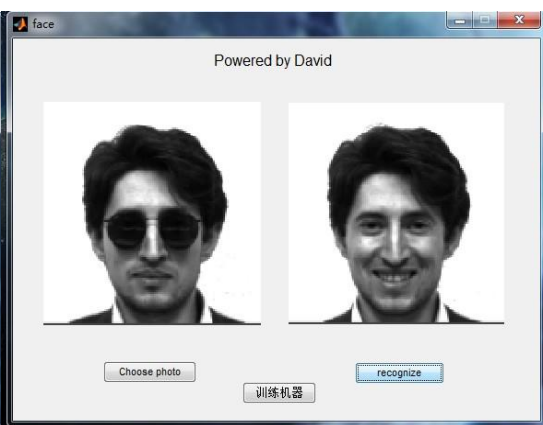


图 4 识别后

由于利用了标准库，并且识别的人数不是很多，也没有选择有大块左阴影和右阴影的人作为训练集以及测试，所以最终的结果还是非常不错的，识别率可达 100%。

但是选择有较大阴影的人做测试，则会出现识别错误，所以 PCA 算法还是存在一定的局限性。



图 5 识别错误

附录——matlab 源码

```
function pushbutton1_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% read image to be recognize
global im;
[filename, pathname] = uigetfile({'*.bmp'}, 'choose photo');
str = [pathname, filename];
im = imread(str);
axes(handles.axes1);
imshow(im);

% --- Executes on button press in pushbutton2.
function pushbutton2_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

global im
global reference
global W
global imgmean
global col_of_data
```

```

global pathname
global img_path_list

% 预处理新数据
im = double(im(:));
objectone = W'*(im - imgmean);
distance = 100000000;

% 最小距离法，寻找和待识别图片最为接近的训练图片
for k = 1:col_of_data
    temp = norm(objectone - reference(:,k));
    if(distance>temp)
        aimone = k;
        distance = temp;
        aimpath = strcat(pathname, '/', img_path_list(aimone).name);
        axes( handles.axes2 )
        imshow(aimpath)
    end
end

% 显示测试结果
% aimpath = strcat(pathname, '/', img_path_list(aimone).name);
% axes( handles.axes2 )
% imshow(aimpath)

% --- Executes on button press in pushbutton3.
function pushbutton3_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

global reference
global W
global imgmean
global col_of_data
global pathname
global img_path_list

% 批量读取指定文件夹下的图片 128*128
pathname = uigetdir;
img_path_list = dir(strcat(pathname, '\*.bmp'));
img_num = length(img_path_list);
imagedata = [];

```

```

if img_num > 0
    for j = 1:img_num
        img_name = img_path_list(j).name;
        temp = imread(strcat(pathname, '/', img_name));
        temp = double(temp(:));
        imagedata = [imagedata, temp];
    end
end
col_of_data = size(imagedata,2);

% 中心化 & 计算协方差矩阵
imgmean = mean(imagedata,2);
for i = 1:col_of_data
    imagedata(:,i) = imagedata(:,i) - imgmean;
end
covMat = imagedata'*imagedata;
[COEFF, latent, explained] = pcacov(covMat);

% 选择构成 95%能量的特征值
i = 1;
proportion = 0;
while(proportion < 95)
    proportion = proportion + explained(i);
    i = i+1;
end
p = i - 1;

% 特征脸
W = imagedata*COEFF;    % N*M 阶
W = W(:,1:p);          % N*p 阶

% 训练样本在新坐标基下的表达矩阵 p*M
reference = W'*imagedata;

```